Project Progress Report

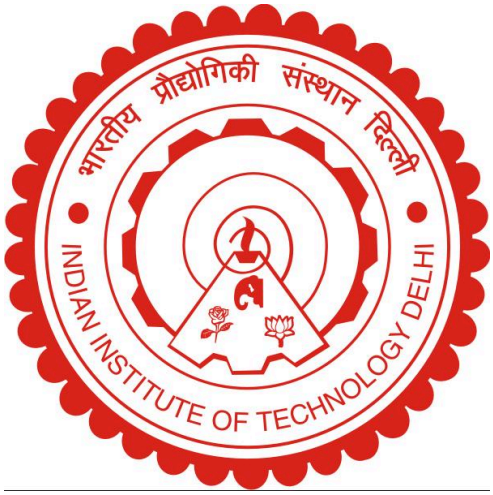
UDP Based Speed Test using Machine Learning

Submitted By :

Manish Kumar (2023MCS2497)

Pratham Agrawal (2023MCS2486)

Submitted On : **21st March, 2025**



DEPARTMENT OF COMPUTER SCIENCE & TECHNOLOGY
Indian Institute of Technology, Delhi
Hauz Khas, New Delhi, 110016

Academic Year: 2024-25

1. Introduction

The key idea of this Project was to use UDP packets to see if using the data provided by UDP packets, can we estimate the **"AVAILABLE BANDWIDTH" (A)** and **"BOTTLENECK CAPACITY" (C).**

2. EXISTING WORK

One of the existing works in this field of study is the paper proposed by Khangura et. al. The paper proposes two Machine learning models:

1. **A Simple NN based Model:**

   The model extracts UDP based data such as rin (The rate (Mbps) at which probe packets are sent from the sender) and rout (The rate (Mbps) at which probe packets are received at the receiver).

   Using these the model realises a new feature called gin (Input gap) (The time gap (seconds) between consecutive probe packets at the sender) and gout (Output gap) (The time gap (seconds) between consecutive probe packets at the receiver.)

   The Relation is as follows:
   gin= l/rin
   gout=l/rout

   Where l is packet size (12112 bits for ethernet frames).

Next Calculation is the gout/gin ratio:
This ratio is a key metric for estimating available bandwidth. It tells us **how much probe packets are affected by cross-traffic**:

- **If gout/gin =1** → No congestion, available bandwidth is **not exceeded**.
- **If gout/gin>1** → Cross-traffic has delayed packets, indicating **available bandwidth is saturated**.
- **If gout/gin$\gg$1** → Severe congestion is occurring.

The Simple NN based model directly estimates the Available Bandwidth A and Bottleneck Capacity C using a feedforward NN.

Traditional bandwidth estimation uses **packet dispersion models** (Pathload, IGI/PTR).

The paper proposes **replacing them with a neural network** that learns patterns from **gout/gin** ratios.

The model takes **multiple probe rate measurements** at once and predicts A and C in a **single forward pass**.

The input is a **k-dimensional vector** of **gout/gin** ratios measured at different probe rates:

X=[gout1/gin1, gout2/gin2 ,… ,goutkgink] Each element represents **how much a specific probe rate was affected by cross-traffic**.

- **Lower values (~1.0)** indicate no congestion.
- **Higher values (>1.2, 1.5, etc.)** indicate congestion.

The **simple neural network** consists of:

1. **Input Layer** → Takes a k-dimensional vector of **gout/gin** ratios.
2. **Hidden Layer** → Fully connected layer with **ReLU activation** (captures patterns in dispersion).
3. **Output Layer** → Two neurons, one for predicting **A** (available bandwidth) and one for **C** (capacity).

2. Iterative Based NN Model:

Instead of performing one-shot estimation, this method iteratively probes the network and refines the bandwidth estimate step by step using two neural networks:

1. Bandwidth Estimation Network → Estimates available bandwidth (A) from previous probe results.
2. Probe Rate Recommender Network → Selects the next probe rate to improve estimation accuracy.

This replaces binary search with an ML-driven approach that learns to efficiently converge to the true available bandwidth**.**

How the Model Works

1. Start with an initial probe rate r0 and send packets.
2. Measure gout/gin and update input vectors.
3. Use the Bandwidth Estimation Network to predict A.
4. Use the Probe Rate Recommender Network to suggest the rnext.
5. Repeat until convergence (i.e., when the bandwidth estimate stabilizes).

Model Architecture

(A) Bandwidth Estimation Neural Network

- Input: A kkk-dimensional vector of past gout/gin ratios
- Hidden Layer: Fully connected layer with ReLU activation.
- Output: A single neuron predicting A (available bandwidth).

(B) Probe Rate Recommender Neural Network

- Input: A k-dimensional vector of past probe rates.
- Hidden Layer: Fully connected layer with ReLU activation.
- Output: A single neuron suggesting rnext.

3. Current Progress:

The current progress done by us, has been the estimation of gout/gin ratios. Based on those ratios we also have built the **Simple NN** model proposed by the paper, and are currently working on building the iterative based model which shall be complete soon.

The code for the same can be found on Github.

The model currently produces these scores at K=25:

Test Loss: 0.10668281465768814
Test MAE: 0.2569030225276947

4. Next Steps:

The next step is to build the Iterative NN model proposed by the paper, after which we shall try to do an extensive feature engineering in an attempt to improve the models scores, we shall also attempt if we are able to produce **similar** scores at a lower computation cost, resulting in a more efficient model compared to the proposed model.